

2.4.2 Program Testing

May/June 2003

3. (a) Describe each of the following types of program error, giving an example in each case.

- (i) Syntax error.
- (ii) Logic error.
- (iii) Arithmetic error.

[6]

(b) Describe two methods used to assist in finding program errors.

[4]

May/June 2004

3 (b) Explain what is meant by the following types of programming error. Give an example in each case.

- (i) Syntax.
- (ii) Logic.
- (iii) Arithmetic.

[6]

May/June 2008

2. (c) When a computer runs a program, the program may fail to run successfully because there are errors in the code.

Describe two types of error that may be present, giving an example of each.

[6]

Oct/NOV 2010. P11/P12

2. (b) State what is meant by the following types of programming error:

- (i) syntax error
- (ii) arithmetic error

[1]

[1]

Oct/NOV 2010. P13

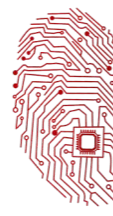
2 (b) State what is meant by the following types of programming error:

- (i) syntax error
- (ii) logic error

[1]

[1]





2.4.2 Program Testing

May/June 2011. P23

1 Jodelle needs to write program code that will check the password to her personal computer.

It checks each attempt to enter the password and closes the screen after three wrong attempts.

She wants the log-in screen to display:

- a request to enter the password
- space to enter the password
- how many attempts have been made
- a message if the log-in has been unsuccessful
- a means of returning to the previous screen

Jodelle first produces her solution using pseudocode. She wants the password to be 'poppy', the name of her cat.

Attempt ← 1

REPEAT

 INPUT Password

 Attempt ← Attempt +1

UNTIL Password = "poppy" OR Attempt = 3

IF Password = "poppy"

 THEN

 OUTPUT "password correct"

 ELSE

 OUTPUT "no valid password entered"

ENDIF

(d) There is an error in this pseudocode.

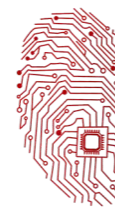
- One way to correct this is to change the first line. Rewrite this line to make the pseudocode correct. [1]
- State the type of error that Jodelle made. [1]

Oct/NOV 2011. P21

1 Ahmed is writing a program to record the data of members of the school football squad.

(g) The squad has 30 players. Ahmed stores the records in an array called Squad. To calculate how many players are defenders he designs this pseudocode:





2.4.2 Program Testing

```
Dtotal ← 0
ArrayPosition ← 1
REPEAT
    IF Squad[ArrayPosition].Position = 'd'
        THEN
            Dtotal ← Dtotal + 1
        ENDIF
    ArrayPosition ← ArrayPosition + 1
UNTIL ArrayPosition = 30
```

This pseudocode will only consider the first 29 records in the array.

- (i) State the name of this type of error. [1]
- (ii) State the line that needs changing. [1]
- (iii) Re-write this line to ensure the pseudocode will consider all 30 records. [1]

May/June 2012. P21/22

2 Philipe is trying different ways of designing the process of entering data into an array.

He declares a variable called `ArraySize` and sets it to 3.

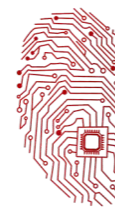
He declares an array `Number[ArraySize]`.

He then writes the following pseudocode.

```
Element ← 1
WHILE Element < ArraySize DO
    INPUT Number[Element]
    Element ← Element + 1
ENDWHILE
```

- (b) (i) There appears to be an error in the above pseudocode. State the type of error. [1]
- (ii) The error can be corrected by changing one line. Write the corrected line of pseudocode. [1]





2.4.2 Program Testing

May/June 2012. P23

3 Liliane wants to write a program to play chess. She will represent the board of 8 x 8 squares, using the 2-dimensional array `Board[8, 8]`.

Each element of the array will need initialising to zero. Later, if a chess piece is on a square, it will take a value of 1.

She starts by writing pseudocode for the initialisation of a 4 x 4 board. This is easier to trace.

```
01 RowNo ← 1
02 WHILE RowNo < 4 DO
03     ColumnNo ← 1
04     WHILE ColumnNo < 4 DO
05         Board[RowNo, ColumnNo] ← 0
06         ColumnNo ← ColumnNo + 1
07     ENDWHILE
08     RowNo ← RowNo + 1
09 ENDWHILE
```

(b) There is an error in the pseudocode.

State the type of error.

[1]

May/June 2013. P21/22

4 (b) Before making the program available it must be tested.

- (i) State when and how syntax errors are detected.
- (ii) State when and how logic errors are detected.

[2]

[2]

May/June 2013. P23

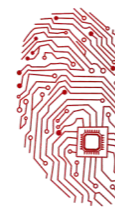
4 (c) Meena has compiled the program and she thinks it is working.

What two types of error could still occur in the program?

For each type give an example.

[4]





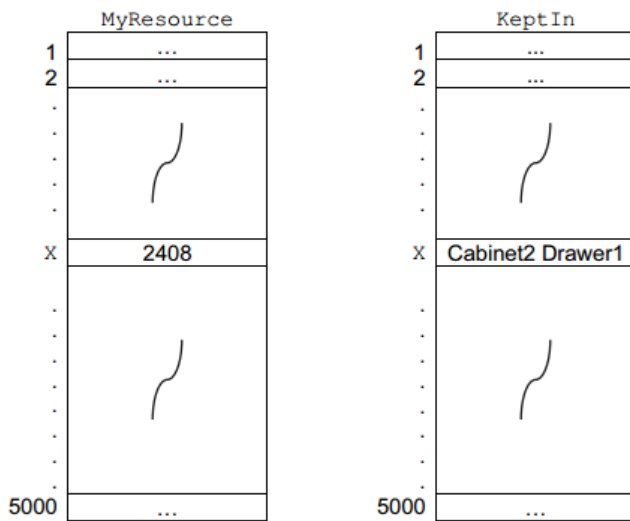
2.4.2 Program Testing

Oct/Nov 2013.P21

3 An array, `MyResource`, size 5000, data type `INTEGER`, is used to store the resource IDs.

An array, `KeptIn`, size 5000, data type `STRING`, is used to store where a resource is kept.

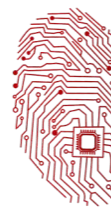
A resource with resource ID `MyResource[X]` is kept at `KeptIn[X]`, where `X` is an integer variable.



Juan writes the pseudocode that searches `MyResource` for a given resource ID and outputs where the resource is kept.

```
flag ← 0
INPUT P
FOR X ← 1 TO 5000
  IF myresource[X] = P
  THEN
    OUTPUT keptin[X]
  flag ← 1
ENDIF
NEXT
IF flag = 0
THEN
  OUTPUT "Not Found"
ENDIF
```





2.4.2 Program Testing

(e) (i) If Juan were to perform a dry run on the pseudocode using a trace table, state one type of error he might find. Give an example.

State another type of error he might find later. Give an example.

[4]

Oct/Nov 2013.P22

3 Aisha wants to write a program that checks the password to her personal computer. The program should check each attempt to enter the password correctly and should terminate after three wrong attempts.

She wants the log-in screen to display:

- a prompt to enter the password
- space to enter the password
- how many attempts have been made
- if the log-in has been successful or not
- a means of cancelling the log-in process

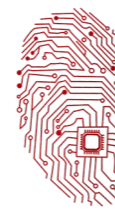
Aisha writes her first try at designing the code in pseudocode. She wants the password to be "Aisha", her name.

```
1 Attempt ← 0
2 REPEAT
3     INPUT Password
4     Attempt ← Attempt + 1
5 UNTIL (Password = "Aisha") OR (Attempt > 3)
6 IF Password = "Aisha"
7     THEN
8         OUTPUT "Password correct"
9     ELSE
10        OUTPUT "No valid password entered"
11 ENDIF
```

(c) This piece of code does not do what Aisha intended. There is an error.

- (i) State the type of error. [1]
- (ii) There are several ways to correct this. One is to change line 5. Rewrite line 5. [1]





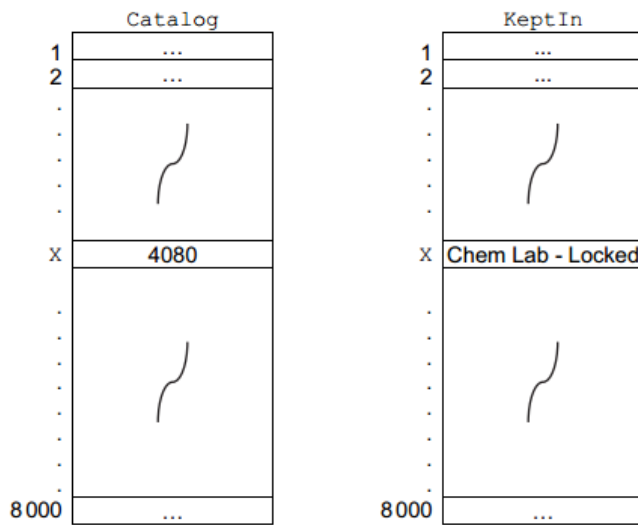
2.4.2 Program Testing

Oct/Nov 2013.P23

3 An array, *Catalog*, size 8 000, data type *INTEGER*, is used to store the equipment IDs.

An array, *KeptIn*, size 8 000, data type *STRING*, is used to store where a piece of equipment is kept.

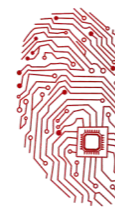
A piece of equipment with equipment ID *Catalog*[*X*] is kept at *KeptIn*[*X*] where *X* is an integer variable.



Ashvin writes the pseudocode that searches *Catalog* for a given equipment ID. The output is to be where the piece of equipment is kept.

```
flag ← 0
INPUT P
FOR X ← 1 TO 8 000
  IF catalog[X] = P
  THEN
    OUTPUT keptin[X]
  flag ← 1
  ENDF
NEXT
IF flag = 0
  THEN
    OUTPUT "Not Found"
  ENDF
```





2.4.2 Program Testing

(e) Ashvin will perform dry runs on the pseudocode using a trace table and a variety of test data. He hopes to find any logic errors and avoid run-time errors occurring later.

- (i) State what is meant by a logic error and give an example. [2]
- (ii) State what is meant by a run-time error and give an example. [2]

May/June 2015.P21/P22

2 The pseudocode below is intended to calculate the sum of a sequence of integers input.

The dummy value -1 ends the input.

```
DECLARE x : INTEGER
```

```
DECLARE Result : INTEGER
```

```
x ← 0
```

```
Result ← 0
```

```
WHILE x <> -1
```

```
    INPUT x
```

```
    Result ← Result + x
```

```
ENDWHILE
```

```
OUTPUT Result
```

(a) (i) The sequence of numbers 3, 5, 2, 1 is input and terminated with -1. Complete the trace table.

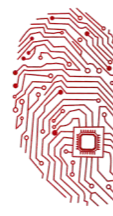
x	Result	x <> -1
0	0	

[4]

(ii) Give the expected result from the sum of the numbers 3, 5, 2, 1. [1]

(iii) What is the error in the given pseudocode? [1]





2.4.2 Program Testing

(iv) State the type of error. [1]

(b) Rewrite the pseudocode so that it works correctly. [3]

Oct/Nov 2015.P21/P23

2 (b) Alia writes pseudocode to convert a coded number using

- the function `DenaryDigit()` from part (a)
- the string manipulation functions `MID()` and `LENGTH()`

Study the pseudocode:

```
01 PROCEDURE ConvertToDenary(CodedNumber : STRING)
02
03     Denary ← 0
04
05     FOR i ← 1 TO LENGTH(CodedNumber)
06         ThisChar ← MID(CodedNumber, i, 1)
07         ThisNumber ← DenaryDigit(ThisChar)
08         Denary ← Denary + (ThisNumber * 10)
09     ENDFOR
10
11     OUTPUT Denary
12 ENDPROCEDURE
```

(iii) State the type of error Alia made in her pseudocode. [1]

(iv) There are two other types of error that can occur when writing or executing program code. [6]

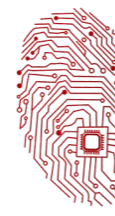
Name each type of error and describe when and how it is detected.

Oct/Nov 2015.P22

2 (f) Ali has written the final program code. There are syntax and logic errors in his program.

- (i) Describe when and how syntax errors are detected. [2]
- (ii) Describe when and how logic errors are detected. [2]





2.4.2 Program Testing

Computer Science (9608)

May/ June 2016. P21/P22

6 A string-handling function has been developed. The pseudocode for this function is shown below. For the built-in functions list, refer to the **Appendix** on page 18.

```
FUNCTION SSM(String1, String2 : STRING) RETURNS INTEGER
DECLARE n, f, x, y : INTEGER
n ← 0
f ← 0
REPEAT
    n ← n + 1
    x ← n
    y ← 1
    WHILE MID(String1, x, 1) = MID(String2, y, 1)
        IF y = LENGTH(String2)
            THEN
                F ← n
            ELSE
                x ← x + 1
                y ← y + 1
            ENDIF
    ENDWHILE
UNTIL (n = LENGTH(String1)) OR (f <> 0)
RETURN f
ENDFUNCTION
```

(iii) There is a problem with the logic of the pseudocode. This could generate a run-time error. Describe the problem.

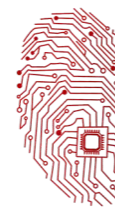
[2]

Oct/Nov 2016. P22

5 A team keeps a record of the scores made by each of their eight players in a number of games. The data in the two tables below shows:

- the scores of the eight players after twenty games
- the eight player names.





2.4.2 Program Testing

	1	2	3	8
1	12	17	67	31
2	35	82	44	29
3	61	39	80	17
4	81	103	21	11
5	56	0	98	4
...				
19	45	6	81	77
20	12	11	3	6

1	Vorma
2	Ravi
3	Chada
4	Nigam
5	Bahri
6	Smith
7	Goyal
8	Lata

The team wants a computer program to input and record the player data.

(c) The team wants the program to produce a report, with the following specification.

The program outputs the total number of player scores that are:

- 50 and over but less than 100
- 100 or higher.

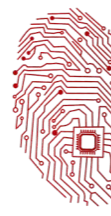
You can assume that before the section runs, the program has assigned all eight player scores to the `PlayerScore` data structure.

A first attempt at the pseudocode is shown below:

```

01 Total50 ← 0
02 Total100 ← 0
03 FOR PlayerIndex ← 1 TO 8
04     FOR GameIndex ← 1 TO 20
05         IF PlayerScore[GameIndex, PlayerIndex] > 100
06             THEN
07                 Total100 ← Total100 + 1
08             ELSE
    
```





2.4.2 Program Testing

```
09             IF PlayerScore[GameIndex, PlayerIndex] > 50
10                 THEN
11                     Total150 ← Total150 + GameIndex
12             ENDIF
13     ENDIF
14 ENDFOR
15 ENDFOR
16 OUTPUT Total150
17 OUTPUT Total100
```

(ii) Consider the following two statements.

Write either TRUE or FALSE next to each statement.

Statement	TRUE or FALSE
The pseudocode considers all the scores for a player, before progressing to the next player.	
The pseudocode considers all scores in a game, before progressing to the next game.	

[1]

(iii) The programmer has made logic errors in the design.

State a line number at which an error occurs.

Explain the error or write the corrected pseudocode statement.

[1]

